REMARKS

This amendment is submitted in response to the Examiner's Action dated August 23, 2003. Applicants have amended the claims to more precisely recite the elements of the invention and overcome the § 112 rejections. No new matter has been added, and the amendments place the claims in better condition for allowance. Applicants respectfully request entry of the amendments to the claims. The discussion/arguments provided below reference the claims in their amended form.

CLAIMS REJECTIONS UNDER 35 U.S.C. § 112

At paragraph 3 of the Office Action, Claims 18-23 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. Applicants have amended the rejected claims correcting the dependencies, where appropriate, to make the claims definite. The amendments overcome the § 112 rejection, and Applicants respectfully request reconsideration of the rejection in light of the amendments.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103

In the present Office Action, Claims 1-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over *McKay et al.* (U.S. Patent No. 6,138,247) in view of *Morikawa* (U.S. Patent No. 5,898,829). The combination of *McKay* and *Morikawa* does not render Applicants' claimed invention unpatentable because that combination does not suggest the subject matter recited by Applicants' claims.

Applicants' invention provides a method for dynamically replacing a failing operating processor with an idle (inactive) spare processor in a multiprocessor (MP) system (Claim 1). The MP is provided a processor register with settable bits, each corresponding to a processor and utilized to determine/indicate whether the corresponding processor is currently active or inactive (Claims 2-3). Applicants' invention also provides a system management interrupt (SMI) handler to temporarily take control of the operating system (OS) and update system configuration parameters when a spare processor needs to be activated (Claims 9-10).

Applicants' claims specifically recite:

(1) "said spare processor is not allocated any processing load by the operating system (OS)..." (Claim 1).

(2) "said MP includes ... a processor register ..., which indicates which processors among said operating processors and said spare processors are currently available to said OS" (Claim 2)

(3) "setting a bit ... to a first value corresponding to assigning ... operating processors to an active state ... available for allocating load; and setting a next bit ... to a second value corresponding to assigning said spare processor to an inactive state" (Claims 2 - 3); and

(4) "activating a system management interrupt (SMI) when said failing processor is determined to be failing, wherein said SMI messages said OS..." (Claim 9).

McKay generally provides a method for enabling an active spare processor to continuously monitor a MP system having a CompactPCI bus in order to determine whether an operating processor is failing (col. 7, lines 41-43 and col. 8, lines 21-22). *McKay* teaches that the spare processor temporarily takes control of the OS and changes the state of a special arbiter to update system configuration parameters when a failure is detected in the active system processor (abstract, col. 3, lines 38-39 and col. 7, lines 44-50).

The specific sections referenced by Examiner, namely col. 2, lines 10-30 provide: "...a standby system processor **determines** a failure ... **places a special arbiter** in a one master mode ... **determines** that a device is at risk of performing a destructive action, the standby system processor **quiesces** the device...then **places the special arbiter** in a multiple master mode." Col. 3, lines 30-53 describe similar functions of the standby processor, where "[i]f there is a failure in the active system processor, a standby processor takes over control of the devices in a standby mode."

It is clear that *McKay*'s "standby system processor" is actively completing some processing load, (even if in the background), which load includes completion of the above functions indicated in bold font (i.e., determines, places, quiesces, places). These functions cannot be completed when the standby processor is "inactive", which is the initial state of Applicants' spare processor following power on and until one of the other processors fail. Applicants' claimed invention specifically provides an inactive (non-functioning) processor that

does not process any workload (se number 1 above). Unlike *McKay*'s use of a functioning, standby processor to complete the workload associated with monitoring of the active processor, Applicants' claimed invention specifically recites: "holding off a spare processor..., wherein ... said spare processor is **not allocated any processing load**" *(emphasis added)*. Notably